What is claimed is:

1.   A system, said system including a host environment and network interface for receiving network traffic units, the system comprising:

a queue for queuing a first network traffic unit received from a first origin;

5   a buffer for storing a second network traffic unit received from a second origin;

a comparator coupled to said queue and said buffer for comparing said first and second origins to determine whether said first and second origins differ; and

a queue manager for queuing said second unit in the queue if said origins differ, else interrupting a host processor for said second unit and subsequent network traffic

10   units received from said second origin.

2.   A system, according to claim 1, further comprising:

a timer for monitoring elapsed time; and

a unit counter for counting units of the received network traffic;

15   wherein after a predetermined passage of time, or after a predetermined number of traffic units have been received, the host processor is interrupted to process queued network traffic.

3.   A system according to claim 2, wherein the host processor is interrupted

20   to process queued network traffic if a predetermined number of network traffic units are received from a particular origin.

4.   A system according to claim 2, in which the network interface is a network interface card (NIC) having a NIC processor, and wherein receipt of each unit of

25   network traffic causes an interrupt of the NIC processor.

5.   A system according to claim 2, in which the network interface is a network interface card (NIC), and wherein receipt the NIC is polled to determine network traffic.

30   6.   A system according to claim 4, further comprising:

a first memory storing a driver for the network interface card, said driver facilitating communication between the host environment and the network interface card; and

a second memory for storing received network traffic for processing by the host

5 environment;

wherein network traffic stored in said second memory is processed by the host environment on interrupt of the host processor.

7. A system according to claim 5, wherein the second memory is either
10 apportioned from host environment memory, or located on the NIC.

8. A system for reducing interrupts presented to a host environment due to receipt of network traffic by a network interface, said traffic formatted into units, the system comprising:

15 a timer for monitoring elapsed time of a received network traffic;

a unit counting arrangement for counting units of received network traffic; and

a source-address tracking arrangement for identifying whether a particular source address is associated with a significant portion of said network traffic; and

an interrupt suppression arrangement for suppressing interrupts to the host, on
20 receipt of network traffic, until a condition is reached, said condition is selected from a set of conditions consisting of:

a predetermined amount of elapsed time measured by the timer,

a predetermined number of counted units of network traffic counted by the unit counting arrangement, and

25 a predetermined number of network traffic units being received from a particular source address as determined by the source-address tracking arrangement.

9. A method for reducing interrupting a host processor of a host environment on receipt of network traffic units, said interrupting directing the host environment to
30 process received network traffic units, the method comprising:

queuing first network traffic units without interrupting the host processor;

monitoring source-addresses of the first network traffic units; and

if plural first network traffic units have a same origin, then interrupting the host processor on receipt of second network traffic units having the same origin.

5 10. A method according to claim 9, further comprising:

monitoring source-addresses of the second network traffic units; and

if plural second network traffic units have a different origin, then queuing said units without interrupting the host processor;

executing a repeating timeout loop; and

10 interrupting the host processor on expiration of said loop.

  11. A method according to claim 10, in which network traffic units are received by a network interface having a local interface memory, the method further comprising:

storing queued network traffic units in the local interface memory;

15 copying the local interface memory to a system memory within the host environment; and

processing by the host environment of network traffic units in the system memory.

20 12. A method according to claim 9, in which network traffic units are received by a network interface having a local interface memory, the method further comprising:

storing queued network traffic units in the local interface memory;

copying the local interface memory to a system memory within the host environment; and

25 processing by the host environment of network traffic units in the system memory.

  13. A method according to claim 12, in which the network interface includes an interface processor, further comprising:

30 processing the repeating timeout loop with the interface processor; and

interrupting processing the repeating timeout loop on receipt of each network traffic unit; and

performing the receiving and copying of first network traffic units asynchronously to operation of the host processor.

5

14. A method according to claim 12, the method further comprising:

counting queued network traffic units; and

if said counting exceeds a threshold, interrupting the host processor.

10

15. A method according to claim 12, further comprising:

assigning a call-back interrupt to a data transfer function, said transfer function to transfer network traffic units in the local interface memory to the system memory;

continuing executing the repeating timeout loop until interrupted by the call-back interrupt; and

15

directing the host environment to process received network traffic within the system memory.

16. A method, comprising:

queuing a first network traffic unit having a first origin;

20

receiving a second network traffic unit having a second origin;

comparing said first and second origins;

if said first and second origins differ, then queuing said second unit; and

but, if said first and second origins equate, then interrupting a host processor for said second unit and subsequent network traffic units received from said second origin;

25

wherein queuing network traffic units includes storing said units in a memory without interrupting the host processor.

17. A method according to claim 16, further comprising:

executing a repeating timeout loop;

30

counting queued network traffic units;

interrupting the host processor on timeout of said loop to process each queued network traffic unit; and

interrupting the host processor if said counting exceeds a predetermined threshold.

5

18.   An article of manufacture comprising a computing-device readable medium having encoded thereon instructions to direct a processor to perform the operations of:

queuing a first network traffic unit having a first origin;

10   receiving a second network traffic unit having a second origin;

comparing said first and second origins;

if said first and second origins differ, then queuing said second unit;

but, if said first and second origins equate, then interrupting a host processor for said second unit and subsequent network traffic units received from said second origin;

15   wherein queuing a network traffic unit includes storing said unit in a memory without interrupting the host processor.

19.   An article of manufacture according to claim 18, further comprising instructions to perform the operations of:

20   executing a repeating timeout loop;

counting queued network traffic units;

interrupting the host processor on timeout of said loop to process each queued network traffic unit; and

interrupting the host processor if said counting exceeds a predetermined

25   threshold.